1. Conceptos básicos
    1. ¿Qué son las bases de datos?

¿Qué son las bases de datos?

Las bases de datos son información almacenada en ficheros

El Glosario IEEE de Ingeniería del Software (IEEE 1990) define el término base de datos de la siguiente forma.

base de datos: Una colección de datos interrelacionados almacenados conjuntamente en uno o más ficheros de computadora.

Desde este punto de vista, un ejemplo de base de datos muy simple podría ser un sencillo fichero de texto que guardase los datos de los clientes de una empresa. El siguiente es un fragmento posible, pongamos, por ejemplo, almacenado en un fichero de nombre `clientes.txt`:

```
...

Pérez, Juan – C/. Gran Vía, 32 3ºA – 28007 Madrid – 12/02/1970

Sánchez, Juan – Avda. Constitución, 2  Bajo A – Madrid 28007 - 12/10/1940

Joaquín González  / C/ Pez Volador 57 6ºA / 28007 Madrid / 1/3/1969

...
```

En la terminología habitual de ficheros, cada fragmento de información que contiene datos de una misma entidad se denominaría **registro**. En el ejemplo, cada registro ocupa una línea del fichero y guarda la información de un cliente.

Dentro de cada registro, las diferentes informaciones son los valores de los **campos**. En el ejemplo podemos identificar como campos el “nombre del cliente”, la “dirección” y el “código postal y la localidad”, aunque no hay nada explícito en el fichero que nos indique que esos campos son necesarios en cada registro. No obstante, a simple vista parece que esa es la

información registrada. Además, aparece una fecha que podría quizá ser la fecha de nacimiento (o quizá no, solo mirando el fichero es difícil decirlo), aunque no está especificado así directamente en el fichero.

En el fichero anterior se está guardando información de una sola "entidad": los **clientes**. En la mayoría de las bases de datos no se guarda información de una sola entidad, sino de varias. Por ejemplo, además de los clientes podríamos tener otro fichero guardando información de los productos (`productos.txt`) y por ejemplo, información de los pedidos (`pedidos.txt`), que debería hacer referencia a ciertos clientes y ciertos productos que estén almacenados en los otros ficheros. Así, la información en los ficheros está **interrelacionada**, en este caso, la información sobre los pedidos **depende** de la información que hay en los otros ficheros.

Mejor dicho, las bases de datos son información almacenada de acuerdo a una estructura sistemática

El ejemplo anterior puede considerarse una base de datos. No obstante, en la práctica es raro encontrar que se utilicen ficheros de texto para guardar bases de datos excepto para casos muy simples. Con la evolución de la informática, se han desarrollado ciertos **modelos de bases de datos** que hacen más uniforme, eficiente y seguro el almacenamiento de la información. Para entender el concepto de modelo de bases de datos, nos fijaremos en algunos aspectos del fichero anterior, concretamente en los siguientes:

- **¿Cómo se define el formato de los datos?** En este caso, la definición del formato ha sido una decisión del creador de los ficheros, adoptando ciertas convenciones en el orden de los campos. Además, parece que falta consistencia en la definición de los separadores de los campos (en unos registros se usa un guión y en otros una barra).
- **¿Cómo se consulta (o actualiza) la información en esos ficheros?** Dado que la estructura es ad hoc, habrá que escribir programas que "sepan manejar" esa estructura.
- Los ficheros con formatos ad hoc no poseen un formato de datos común, ni formas de hacer consultas o actualizaciones generales. Precisamente para conseguir esos formatos y lenguajes de consulta

comunes se han desarrollado los denominados “modelos de bases de datos”. *modelo de bases de datos: una definición de una forma de estructurar las bases de datos, junto a las operaciones (consultas y actualizaciones) que se pueden hacer sobre esa estructura. Un modelo de bases de datos es teórico, no es un software concreto.*Por ejemplo según el “modelo relacional de bases de datos” los datos se estructuran en tablas que tienen la forma matemática de una relación, y se definen una serie de operaciones matemáticas (selecciones, proyecciones, etc.) para consultarlos. No en todos los casos los modelos de bases de datos tienen una formulación matemática, pero sí tienen una formulación de carácter formal.

Por tanto, el principal inconveniente que tienen los ficheros como el que hemos visto es que la forma de guardar los datos (el orden, la estructura) sigue solamente el criterio del individuo que creó el fichero con los datos. En el ejemplo anterior, en unos casos se usan guiones y en otras barras para separar los campos dentro de un registro, no hay uniformidad en el orden de apellidos y nombre de los individuos, etc. Así, si otras personas (u otras aplicaciones informáticas) quieren utilizarlo tienen que “interpretar” el formato que le dio el creador original.

Esto lleva a un concepto de base de datos que incluye en su definición el concepto de estructura sistemática, es decir, los datos se almacenan de acuerdo a convenciones estrictas que deben seguirse.

base de datos: Una colección de datos interrelacionados almacenados conjuntamente en uno o más ficheros de computadora de acuerdo a una organización y estructura sistemática determinada por un modelo de bases de datos.

Un ejemplo: el modelo relacional de bases de datos

Sin entrar en muchos detalles, podemos decir que el modelo relacional se basa en utilizar tablas relacionales para almacenar la información. Así, tomando el ejemplo anterior, tendríamos la definición de una tabla relacional como la siguiente:

`CLIENTES( id , apellidos, nombre, dirección, cp, localidad, fecha-nacimiento)`

En esa relación matemática, cada campo (atributo) estaría definido sobre un domino (un tipo de datos determinado), y el atributo `id` está marcado como la clave primaria, es decir, un dato que identifica unívocamente a cada "fila" (tupla) de la tabla. Como vemos, los tipos de datos y la forma de estructurar la información están definidos por el modelo.

Si quisiésemos hacer una consulta sobre la tabla anterior, no haría falta hacer un programa que tratase con la esturctura de un fichero, sino que utilizaríamos un lenguaje de bases de datos. Concretamente, el lenguaje SQL es un lenguaje estandarizado para bases de datos relacionales. Una consulta SQL que obtiene los clientes de una localidad sería la siguiente:

```
select apellidos, nombre from clientes where localidad = 'Madrid'
```

El anterior es un ejemplo de cómo los modelos de bases de datos, los lenguajes estandarizados y el software que los implementa permite facilitar el desarrollo de aplicaciones. Por un lado, SQL es un lenguaje ampliamente utilizado y conocido, y es fácil encontrar documentación sobre él. Por otro lado, los detalles de cómo se almacenan los datos quedan ocultos tras los lenguajes, permitiendo al programador centrarse en otras tareas, y delegando la gestión de los datos y las consultas al software que soporta esos modelos y lenguajes.

Otro ejemplo: un modelo de bases de datos orientado a objetos

Como contraste con el modelo relacional, es interesante conocer modelos orientados a objetos. Aunque las bases de datos orientadas a objetos no han tenido una implantación tan extendida como las relacionales, si han llegado a contar con estándares. Por ejemplo, el estándar ODMG 3.0 permite definir los datos mediante los propios lenguajes de programación. Así, si utilizamos Java, bastaría con definir una clase como la siguiente:

```
class Cliente{
String apellidos;
String nombre;
Direccion dir;
String cp;
Localidad loc;
java.util.Date fechaNac;
public String getApellidos(){
//...
}
// otros métodos...
}
```

Donde `Direccion` y `Localidad` serían otras clases Java definiendo objetos que representan direcciones, y podemos ver que se utiliza la clase `Date` definida en las bibliotecas de Java. La ventaja de este modelo de bases de datos es que es muy cercano a la representación de los datos que se utiliza en los lenguajes orientados a objetos (de hecho, las definiciones de los datos son las mismas clases que se utilizan en la programación, como la clase `Cliente`), y permite aplicar el diseño orientado a objetos en la medida que queramos.

Para consultar estas bases de datos uno de los lenguajes posibles es el object query languaje (OQL), similar al SQL.

Los Sistemas de Gestión de Bases de Datos implementan software para gestionar datos de acuerdo a un modelo

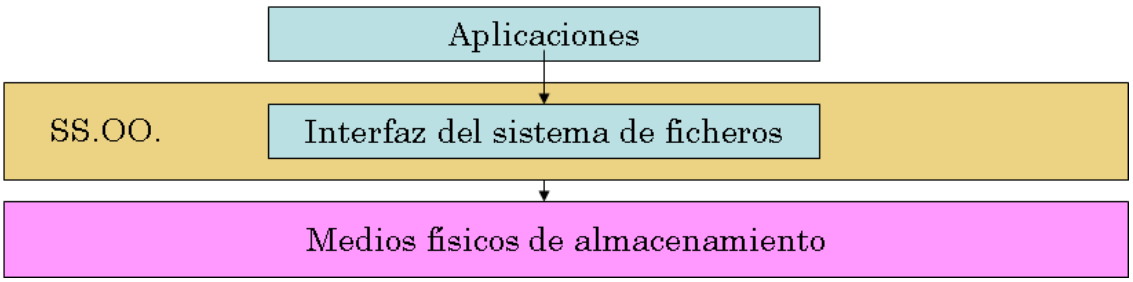
La normalización de la estructura y los lenguajes de acceso permite construir software especializado que se encargue de tratar con los datos en un formato estandarizado. A ese software especializado se le denomina **Sistema Gestor de Bases de Datos** (SGBD). Los SGDB además incluyen muchas funciones como la seguridad y el control de acceso, adicionales a la mera estandarización de las estructuras y los lenguajes de consulta y actualización.

Hay muchas aplicaciones software que cumplen (en mayor o menor medida) con el modelo de bases de datos relacional, desde aplicaciones de ofimática como Base en OpenOffice hasta software complejos con muchas funciones, como el gestor de bases de datos MySQL. Habitualmente, un SGBD se ajusta a un modelo de base de datos, así se dice que es un “gestor relaciona” o un “gestor orientado a objetos”, etc., aunque actualmente hay SGBD que combinan aspectos de varios modelos.

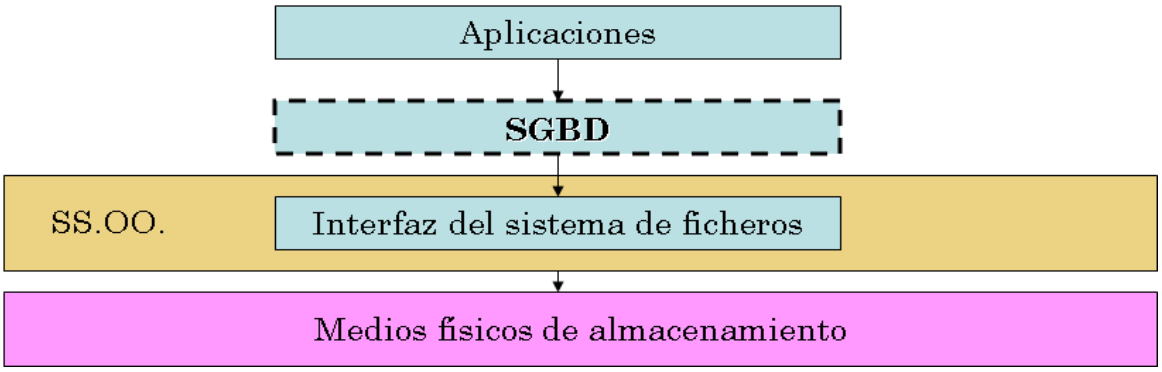
El concepto de Sistema Gestor de Bases de Datos (SGBD) se define en el Glosario IEEE de Ingeniería del Software de la siguiente forma.

Sistema Gestor de Bases de Datos (SGBD). Un sistema informático compuesto por hardware, softwar e o ambos, que proporciona una técnica sistemática para la creación, el almacenamiento, el procesamiento y la consulta de la información almacenada en base de datos. Un SGBD actúa como un intermediario entre las aplicaciones y los datos, o bien entre los datos y la base de datos. […]

Es decir, un SGDB es un software (normalmente el mismo software de SGBD puede instalarse en configuraciones hardware muy variadas) que proporciona funcionalidad añadida al sistema de ficheros para facilitar la gestión de datos. Las Figuras 1 y 2 muestran esta diferencia. En la Figura 1, se tiene una aplicación que directxamente funciona sobre la interfaz de ficheros del sistema operativo, que a su vez gestiona los medios físicos de almacenamiento (discos duros, discos ópticos, etc.).



La Figura 1 representa la gestión de los datos mediante ficheros como el mencionado clientes.txt más arriba.



En la Figura 2 se muestra como el SGBD proporciona funciones adicionales, de modo que las aplicaciones no trabajan directamente con los ficheros y directorios, sino que utilizan funciones de los gestores de bases de datos.

Los ficheros son unidades de almacenamiento básicas de los Sistemas Operativos. Un fichero a ese nivel de abstracción es una secuencia de bytes con un nombre. Las aplicaciones abren y usan los ficheros y la asignación del espacio (contigua, enlazada, con índices) en los medios físicos es transparente al usuario.

El uso de un SGBD proporciona una serie de ventajas que han de valorarse a la hora de seleccionar su uso, que vienen a resolver los problemas que presenta el simple uso de ficheros. Los SGBD han evolucionado hasta convertirse en piezas de software complejas que requieren habilidades de configuración y administración específicas.